

la.isla.minima.2014.ass.md

本篇介绍了：一定程度上，外乡人如何去翻译一门陌生语言的电影文本（生肉），并制作成字幕文件。按照发生顺序，介绍了涉及的软件、插件、命令行等工具。内容主要包括：如何安装，部分设置和操作。

本次作业仅出于个人学习目的，其中所处理的字幕、截图没有授权，只在必要时，作为著作进行引用。

源文件.srt

一些影视DVD的光盘文件，有一定几率掉落母语版字幕，它通常有正确的时间轴。2014年的西班牙剧情、悬疑题材电影《沼泽地》，它的西语版.srt (SubRip Text) 文件少见地通篇带有一些电影脚本内容，包含背景声、音效、人物演出等注释。这些通常是会剔除的部分，比如在它的英译版字幕里。保留这些内容——

这里挪用影片中的一个角色事件「无证狩猎」——先去拿到几个关键道具：电子词典浏览器GoldenDict、网页浏览器插件沙拉查词、Python包autosub、Rust包alass-cli，便开始寻迹和追踪。

翻译

主要步骤：

1. 播放并逐句查看媒体文件。
2. 使用在线翻译服务reverso.net快速输出整句或片段的意思。
3. 使用GoldenDict查找单词的详义。这里用的是西班牙语-英语词典，所以还需使用沙拉查词作中介，翻译英语。
4. 借助母语的读写能力，从译义组织译文。

Scoop

如果网络条件较好，可参考Scoop官网，先安装这个软件包管理器。

1. `Win + Q` > Windows PowerShell > 以管理员身份运行。
2. `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser`。
3. `Invoke-Expression (New-Object System.Net.WebClient).DownloadString('https://get.scoop.sh')`。
4. (可选) 安装一个下载工具，`scoop install aria2`。

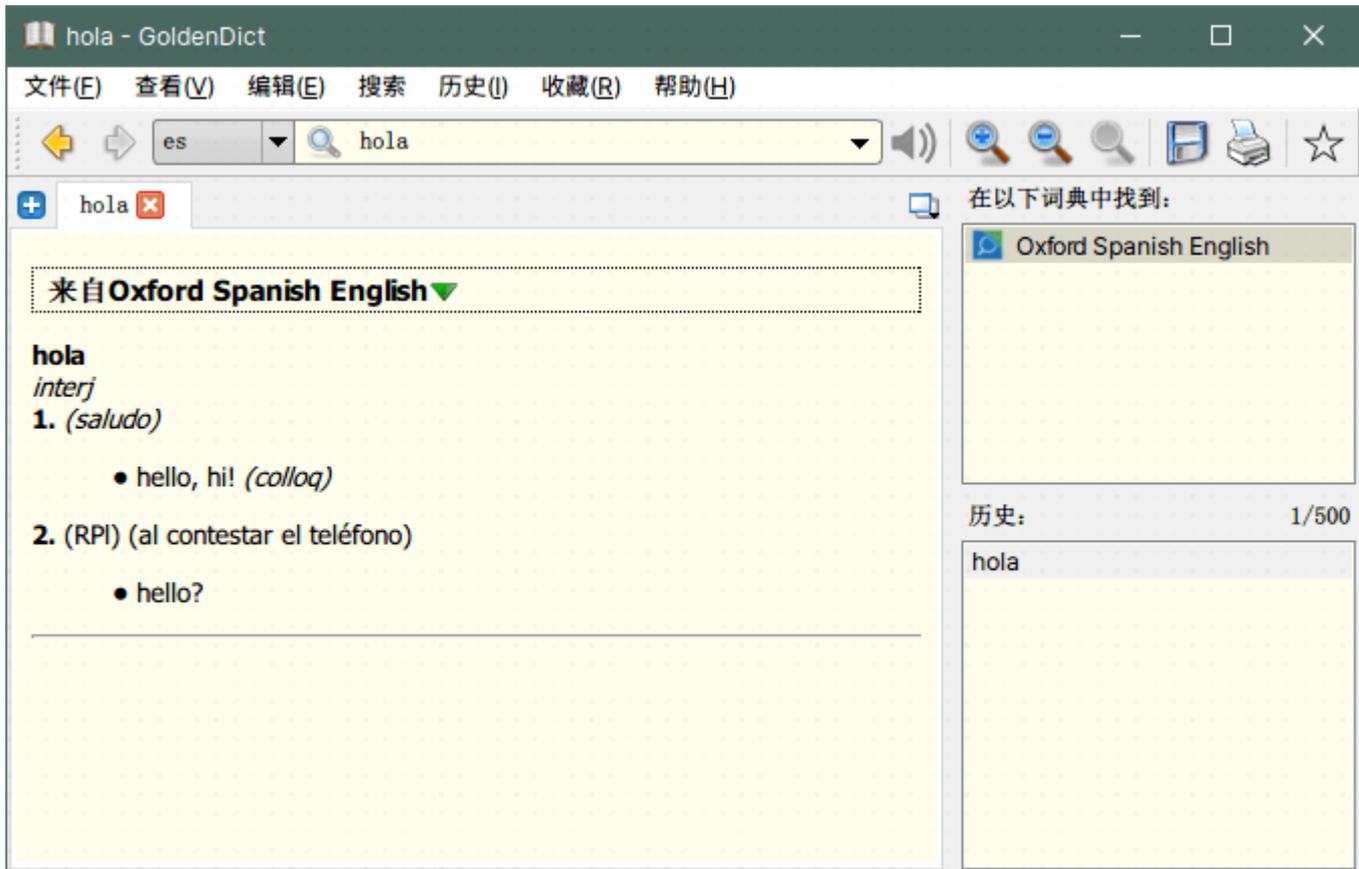
使用时如果遇到报错，可去[此问题页面](#)自查。若需卸载，`scoop uninstall scoop`。

GoldenDict

1. 安装GoldenDict。`scoop install goldendict`，或者到GoldenDict官网下载32-bit Windows installer这个安装包。



2. 下载词典。这里到[掌上百科论坛](#)注册了一个账号，在[该帖子](#)下，下载了Oxford Spanish English（牛津西班牙语-英语词典）这个文件夹——其中的.mdx文件（Mdict词典文件）就是所需的。
3. 导入词典。Win+Q > Goldendict > 在菜单栏选择「编辑」 > 词典 > 词典来源 > 添加 > 选择刚才的文件夹 > 应用 > 等待扫描完成。
4. 回到软件的主窗口，便可以在导航栏搜索词汇了。



编辑词典群组

GoldenDict默认会启用所有词典。当你有多本词典时，就可以给词典编组，以便可以筛选语种。

1. 编辑 > 词典 > 词典来源 > 添加 > 文件夹A > 递归扫描 (On)。这里，你的词典文件夹都在文件夹A下。
2. 编辑 > 词典 > 群组 > 添加群组。
3. 将左栏的可用词典，拖进右栏的群组里 > 应用。
4. 回到主窗口，下拉搜索框的左段，便可以选择群组了。

SalaDict

安装沙拉查词。参考官网的[下载地址](#)，根据所使用的浏览器进行安装。

按需设置[浏览器外划词](#)，这里以Chrome为例：

1. Chrome > 设置 > 高级 > 系统 > 关闭Google Chrome后继续运行后台应用 (On)。
2. 在浏览器内划词，唤出查词窗口 > 打开设置。



浏览器外划词

让沙拉查词查询浏览器外的划词理论上有两种方式

- 通过剪贴板做中转配合全局快捷键（目前实现）
- 通过扩展 API 与本地程序通讯（尚未实现）

不打开浏览器直接召唤沙拉查词

1. 沙拉查词设置中开启「后台保持运行」。
2. 在 Chrome 设置中搜索 `background` 或者



启动即可。

3. 将 Chrome 加入开机启动。

3. 基本选项 > 后台保持运行 (0n) 。
4. 权限管理 > 读取剪切板 (0n) ，写入剪切板 (0n) 。
5. 隐私设置 > 设置快捷键 > 沙拉查词 在独立窗口中搜索剪贴板内容 > `Alt+Z`，全局。这里的键位请按需设置。
6. 在浏览器外划词，`Ctrl+C`，再 `Alt+Z`，便可以唤出独立查词窗口了。
7. 如果要关闭该窗口——可以用Windows快捷键 `Alt+Esc` 切换到上一次操作的窗口。
8. (可选) 快捷查词 > 出现位置 > 下方。
9. (可选) 快捷查词 > 打开独立窗口设置 > 窗口高度 > 500px。

校对（上）

另一个主要的问题是校对。如果有已校版本，会省去不少麻烦（也就没这个章节）。所以先去查一些来源信息（这里实际上是补充）。在这个.srt文件的末尾，会看到注释：

```
981
01:36:10,100 → 01:37:00,100
Subtitulado por quetzalcoat167
para subadictos
```

这两行分别是：上传者的ID，来源或分发的地址。找到此处地址，在[该帖子](#)下，上传者提到：

- 使用了Subtitle Edit对.sup (Subtitle Presentation) 文件进行OCR（光学字符识别），制作了.srt。
- 将一些安达卢西亚 (Andaluz) 方言转写成了官方西班牙语——卡斯蒂利亚 (Castellano) 语，以便于理解。
- 删除了一些音画不同步的提示音乐的字符行。

之后，有所准备地，偶尔就会接触到这类差错：



类型	示例	行号	处理
错别字	Citroén > Citroen	27, 149	查找并替换
听写错误	Déjate de tonterías	622	使用STT工具检查
听写错误	ah, si. > ah, si?	934	参考 电影脚本PDF文件
文本缺失		646-647	参考 英语版字幕
语法错误			忽略

其中，与STT (Speech-to-Text) 相关的步骤是：

1. 使用媒体播放器MPV设置A-B点循环。
2. 用视频录制、直播软件OBS，录制视频片段或音频片段。
3. 使用命令行软件autosub识别片段，输出到字幕文件。

MPV & OBS

`scoop install mpv`，或者[下载MPV安装包](#)。

`scoop install obs-studio`，或者[下载OBS安装包](#)。

1. `Win + Q` > MPV > 拖入媒体文件。
2. 在MPV播放时，按`L`设置A点，再按`L`设置B点，并开始循环。（按第三下`L`则清除A-B循环）
3. `Win + Q` > OBS > 来源 > `加号` > 显示器采集。（或者选音频输入采集）
4. 控件 > 开始录制。（录制时，可循环播放多次）
5. 停止录制。会得到一个以日期和时间命名的录像文件。

设置录像参数

A卡用户可参考[这个视频](#)，设置OBS录制参数：

- 输出
 - 录像
 - 录像格式 > mp4
 - 编码器 > H264/AVC
 - 质量预设 > 质量
 - 速率控制方法 > 恒定QP (CQP)
 - I帧QP/P帧QP > 15~17
 - 音频
 - 音频比特率 > 320
- 视频
 - 常用FPS值 > 48

CMD & Cmdr



命令提示符 (Command Prompt)，也就是可执行文件cmd.exe，是Windows系统默认的命令解释器 (Command-line Interpreter)。后者所解释的语境通常是给定的操作系统或编程语言中的一种——这一类连接用户与系统内核、解释和执行指令、返回结果的，通常称为shell (壳层)。

Cmder，自称是Windows上最好的shell，提供命令行界面CLI (Command-Line Interface)、图形用户界面GUI (Graphical User Interface) 两者。

因为Cmder不适合一笔带过，这里就选择CMD作为应急物品。

autosub

先安装Python语言包。去Python官网下载最新安装包，安装时一直选「下一步」即可。或者：

1. Win + Q > 命令提示符。
2. scoop install python。

安装完成后，可测试命令，pip3。如提示「...不是内部或外部命令，也不是可运行的程序或批处理文件」，可尝试重启电脑。接着：

1. 根据此仓库下的自述，安装依赖项，scoop install ffmpeg。
2. pip3 install autosub3。

进行语音识别前，可能需要改变CMD工作的目录，切换到之前的录像文件所在的文件夹：

1. 先切换所在的硬盘，盘符:。例如切换到E盘，即E:。
2. 再切换所在的文件夹，cd 文件夹。例如切换到文件夹A的文件夹B里，即cd E:/文件夹A/文件夹B。文件夹名不要有空格。

识别语音到字幕文件，autosub -S 源语言 -D 目标语言 -o output.srt 媒体文件。此处的语言代码可在ISO 639-1 (国际标准化组织639号标准第1部分) 中查找。

连接 (concatenate) 一下文件，cat output.srt。

```
E:\ λ autosub -S es -D es -o output.srt input.mp4
Converting speech regions to FLAC files: 100% |#####| Time: 0:00:00
Performing speech recognition: 100% |#####| Time: 0:00:05
Subtitles file created at output.srt

E:\ λ cat output.srt
1
00:00:00,256 --> 00:00:02,048
Quién es el hombre sombrero

2
00:00:02,304 --> 00:00:04,864
De tonterías Marina quién es el hombre de sombrero

3
00:00:05,376 --> 00:00:07,680
De tonterías Marina quién es el hombre de sombrero
```

时间轴



参考[此仓库](#)操作。这里需要用到Rust和它的包管理器Cargo。先安装Rust语言包。一般会提示需要C++

build tools :

1. `scoop install vcredist2019`。或者到[此页面](#)下载Visual Studio 2019生成工具并安装。
2. `Win + Q` > Visual Studio Installer > 已安装 > Visual Studio 2019生成工具 > 修改。
3. 工作负荷 > 桌面应用和移动应用 > 使用C++的桌面开发：
 - MSVC v142 - VS2019 C++... (On)
 - Windows 10 SDK... (On) ([参考该问答贴](#))
4. 重启电脑。

从[此处](#)下载rustup-init.exe并运行。如果遇到网络问题，提示「could not download file from...」，可参考[此文档](#)，执行：

1. `set RUSTUP_DIST_SERVER=https://mirrors.sjtug.sjtu.edu.cn/rust-static`。
2. `set RUSTUP_UPDATE_ROOT=https://mirrors.sjtug.sjtu.edu.cn/rust-static/rustup`。

再运行rustup-init.exe，一路默认即可。若需卸载，`rustup self uninstall`。

安装成功后，`cargo install alass-cli`。

进行自动较轴，`alass-cli 媒体文件 源字幕 目标字幕`。

```
E:\> alass in.mkv in.srt out.srt
extracting audio from reference file 'in.mkv'...
444 / 444 [=====] 100.00 % 7.79/s

Guessing framerate ratio...
6 / 6 [=====] 100.00 % 7.14/s

info: 'reference file FPS/input file FPS' ratio is 1

synchronizing 'in.srt' to reference file 'in.mkv'...
461 / 461 [=====] 100.00 % 527.89/s

shifted block of 461 subtitles with length 1:26:30.860 by 0:00:00.000
```

在仓库所有者的[学士论文](#)里，他说明该工具的工作原理是基于语音活动检测（Voice-Activity-Detectio），也就提示了一种利于加工、自动化等后期作业的字幕文件的书写规范。

校对（下）

来到了一般称作「二校」「三校」的阶段，差不多就是多次检查。这里列举部分段落：





00:03:06处，文字注释feria译成「交游会」。来自一种早年在小地方举办的，供居民交易、游玩的季节性集会，当地人对此的称呼「交流会」。可能类似大都会的嘉年华。



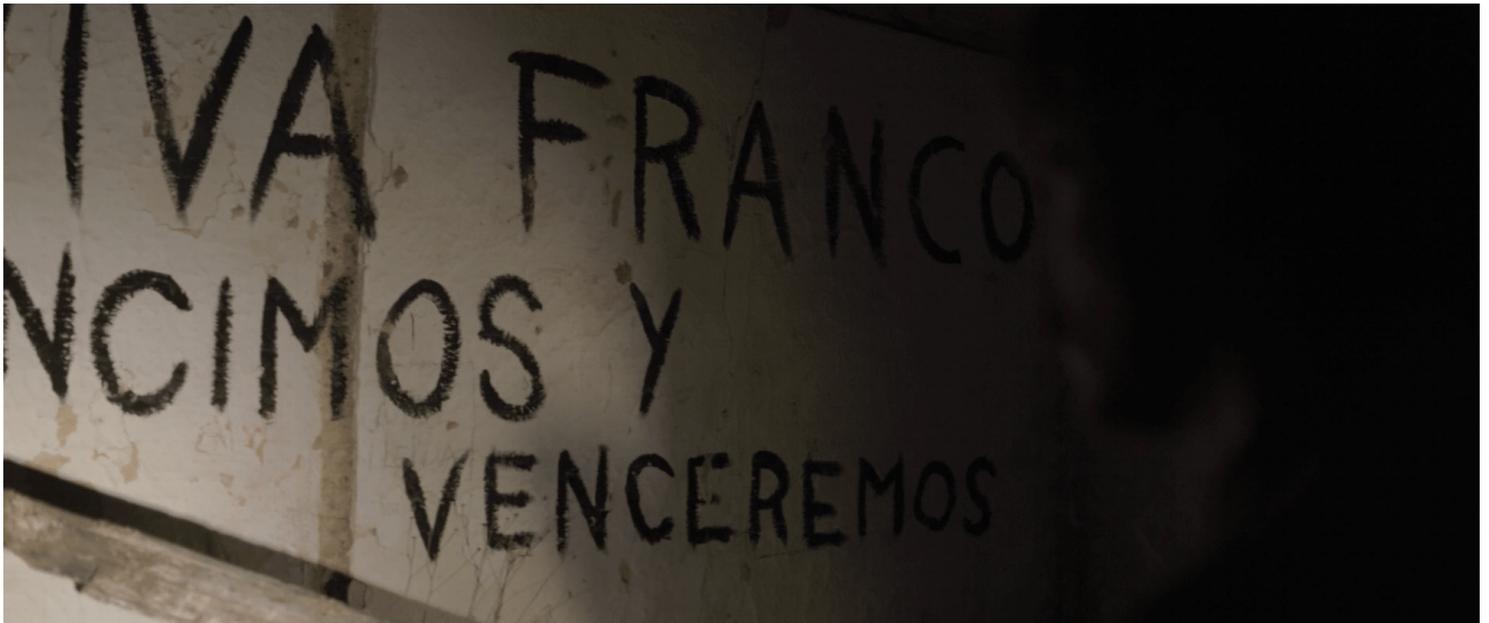
00:05:42至00:06:62，两位主人翁在玩打气枪。这里看机器翻译给的结果，以及参考英语字幕，不能确定胡安说的「现在不是招惹军人的时候」和佩德罗说的「那个法西斯份子」是说哪件事情，和什么人。之前，他们在旅店刚落脚时，在房间的墙上见到了一把纳粹十字——这么一来，佩德罗有可能因此「给报社寄稿」，举报了可能已退役的男掌柜——但事实上，他应该不会在初到小镇的头晚、晚饭与打枪的间隙，找到邮局去寄信。此处确实是交代了佩德罗的人物背景，见该篇[报告4.1](#)处。





00:09:16处，胡安问「这是什么（酒）」，罗西奥回答「cangrejos de rio（河）」，检索出的结果是「小龙虾/淡水螯虾」，而el cangrejo通常又是「螃蟹」。

这里更重视，也将电影自身视为一处语料，也就是在封闭的小范围进行搜索，就能找到——影片内，在01:35:56再次提到了fábrica（工厂） de cangrejos，但在01:07:57附近，画面中出现了一只龙虾。



00:16:08处，涂鸦的人名Franco，在大部分的字典和翻译器里对应「佛朗哥」。这里参考卡斯蒂利亚发音，与Francis/Frances[弗朗西斯]，Francine[弗朗西尼]，Francisco[弗朗西斯科]，Francesca[弗兰西斯卡]等人名音译保持一致，把Franco对应「弗朗哥」。

后期文件.ass

字幕的主体——对白，几类注释——文字、音响、演出（不是专业术语），它们间的时间轴存在重叠，就需要制成.ass（Advanced SubStation Alpha）等格式来保存。涉及到文本编辑器Sublime Text、字幕编辑器Subtitle Edit、特效字幕编辑器Aegisub的使用，这里不再赘述。

thumbnail.jpg

最后，用媒体缩略图工具mt来生成（剥皮）一张4x4格的图：



1. `scoop install go` or open <https://golang.org>
2. or `go env -w GO111MODULE=on`
3. or `go env -w GOPROXY=https://goproxy.cn,direct`
4. `go get github.com/mutschler/mt`
5. `mt -n 16 -c 4 --disable-timestamps --header=false` 媒体文件



相关索引

可能在别的地方再展开介绍：

1. 截图工具 **Snipaste PRO**：截取软件应用的窗口。部分截图在写文字步骤时作参考，选一张图用于展示工具的主界面。
2. 图片压缩工具 **pngquant** 和 **guetzli**：用来代替需要调用 **tinypng** 的 API (Application Programming Interface)，也就是需要联网的 **tinifier**。
3. 文本编辑器 **Obsidian**：以 .md (Markdown) 格式编辑了这篇报告，并且渲染和导出了 .pdf (Portable Document Format) 文件，作为分发的备用路线。
 1. PDF编辑器 **PDF-Xchange Editor**：用来预览 .pdf，并且创建了自签名数字ID，对文档进行数字签名 (Digital Signature)，生成了为 PCCS (Public-Key Cryptography Standards) #12 格式、扩展名为 .pfx 的证书文件。
 2. Golang包 **markpdf**：给 .pdf 的每页添加水印。这里用来铺了一层很浅的底纹，作装饰。
 3. Java应用 **BatchPDFSign**：使用 .pfx 来给 .pdf 签名。
4. 翻译记忆工具 **Omegat**：还没有使用到。据介绍，它建立记忆库来辅助人工翻译，适合于「大型文档、专业术语、衍生案、修订、更新、多语言、共同作业」等。其记忆库文件应该还可以兼容其他的软件，比如本地化工具 **Poedit**。

